NBSIR 73-198

General Purpose Scientific Document Code Operations Under Exec 8

Robert McClenon

Office of Standard Reference Data
National Bureau of Standards
Washington, D. C. 20234

May 1973

Interim Report

Prepared for
Computer Services Division
National Bureau of Standards
Washington, D. C. 20234

NBSIR 73–198

GENERAL PURPOSE SCIENTIFIC DOCUMENT CODE OPERATIONS UNDER EXEC 8

Robert McClenon

Office of Standard Reference Data
National Bureau of Standards
Washington, D. C. 20234

May 1973

Interim Report

Prepared for
Computer Services Division
National Bureau of Standards
Washington, D. C. 20234

U. S. DEPARTMENT OF COMMERCE, Frederick B. Dent, Secretary

NATIONAL BUREAU OF STANDARDS, Richard W. Roberts, Director

Foreword

This report is a manual for users of the General Purpose Scientific Document Code System. It was written to provide guidance during the transition from Exec 2 to Exec 8 on the NBS Univac 1108. A preliminary version was distributed in draft form in October 1972. The present version is only slightly different.

The transition to Exec 8 is nearly complete. Most GPSDC users have learned how to handle common types of runs. But the advice in this manual remains important. The manual tells how GPSDC should be used under Exec 8. New users can use it to learn how. Old users should use it to refresh their memories and expand their knowledge of the system.

The author wishes to thank several people for their assistance in writing this report. Keith Draxl and Blanton C. Duncan read the preliminary versions and commented on them. David Garvin not only read, re-read, criticized and edited the draft, but also contributed the section on the transition from Exec 2 to Exec 8. Special recognition is due to the late Philip J. Walsh, who read the report before its release to check its technical accuracy. The importance of his efforts in assuring a reasonably smooth conversion to Exec 8 cannot be over-estimated. His untimely death deprived the NBS community of computer users of a teacher, a counselor, and a friend.

Contents

General Purpose Scientific Document Code
Operations under Exec 8

by

Robert McClenon

Abstract

Operating procedures for the use of the General Purpose Scientific Document Code text handling system are described. These apply to the use of the system on the NBS 1108 computer under control of the Exec 8 supervisor. Control cards, file definition and handling, compilation of routines and execution of programs are described. Examples of typical runs are given. The report is a manual for users of the system.

Key words: Character codes for scientific text; computerized text processing; NBS computer system; scientific text; users' manual

1. The Transition from Exec 2 to Exec 8

This manual is intended as an introduction to the use of the Document Image Code under the Exec 8 operating system. It is oriented to the user who has a working knowledge of Exec 2 but is less familiar with the more complex Exec 8 system. It is not meant as a substitute either for the complete technical documentation of Exec 8 published by Univac, or for the Exec 8 training seminar offered by the Computer Services Division.

Certain features of Exec 8 will not be discussed in this manual. One of these is the ADD processor, which can be used to insert decks of cards (either data or control) in the run stream. While it is a very useful and powerful tool, it is not mentioned because it had no counterpart in Exec 2. It will be described in the CSD training seminar.

Exec 8 can be used in either the batch processing mode or the demand mode (also known as on-line, conversational, or interactive mode). The GPSDC system, since it was developed under Exec 2, is designed for batch operations. All of the examples in this manual assume batch processing.

As seen by the user the operating programs of the GPSDC system will remain the same after conversion to Exec 8 as they were under Exec 2. (For instance, there will be no change in the "random order free-form data deck." Tapes generated under Exec 2 will be readable under Exec 8.) However, substantial changes will be required in two aspects of job setup. The first is the control cards. In general, the format of Exec 8 control cards is different from that of their Exec 2 counterparts. The letters used to indicate options will also be different in some cases. The second area of difference is recompilation procedures for run-time modification of programs. Typical Exec 2 and Exec 8 run-streams are constrasted below. These examples show how a programmer would make corrections to the subroutine NXTL1 and then run ASKIN (using the corrected NXTL1) to read cards punched in "the world's worst keypunch code" and write the GPSDC output on tape reel 9876, which is assigned as file 3.

Exec 2

```
@B RUN NAMEXY,    12345,1,50,200
@W ASG C=9876
@XQT CUR
 INF DOCIM3
@I FOR,*  NXTL1, NXTL1
   (Correction cards)
@N XQT ASKIN
*OTFILE=3
FILE 3 NEW
*PGMOPT=1
*RM=80
*LF 3=2
*RUN
(Card input)
@EOF
@FIN
```

Exec 8

```
@RUN NAME78,12345-NAMEXY,GPSDC,1,50/200
@ASG,TJ   3,8C,9876
@FOR,S    DICX8.NXTL1,NXTL1,NXTL1
  (Correction cards)
@MAP,N DICX8.ASKIN,ASKIN
@XQT ASKIN
*OTFILE=3
FILE 3 NEW
*PGMOPT=1
*RM=80
*LF 3=2
*RUN
(Card input)
@EOF
@FIN
```

On the other hand, Exec 8 will grant the user greater flexibility in the construction and manipulation of files, both of data and programs. For example the input plus editing cycle can take advantage of the possibility of storing files on mass storage (disc or drum). Input from paper tape, in the present manner, can produce a GPSDC file on mass storage. After it is proofread this file can be edited, perhaps several times, and then the "final" version be put out on tape.

Testing of new programs can also take advantage of this mass storage feature, both for the program decks and the test file used during check out.

Updating of the basic GPSDC file can be done (by those who control the system) when needed, not on a preset schedule as at present.

A basic feature of Exec 8 is "multi-programming". This means that several jobs will be run at the same time, if the requirements for core and I/O units permit. This has a definite impact on our operation. Probably the through-put of jobs during a shift at the NBS Univac 1108 will increase substantially. The turn-around time, on the other hand, may be longer because of the parallel processing. This may be the case for jobs with substantial amounts of input and output. Also, the cost of a job will reflect the amount of core required. GPSDC programs have a large overhead of systems routines. This means that core storage costs may be important. We will have to consider very carefully how to minimize these. One way that will be used from the start will be to make each program into an overlay. This will have an impact on every run-time recompilation.

2. Files and Control Cards

One of the fundamental concepts of Exec 8 is that of a file. A file is simply any set or collection of machine-processable information that is logically treated as one unit by the computer system. A deck of punched data cards read by a program would, in the broad sense, be a file. On the other hand, if two decks of data cards were combined into one, but with control cards separating them, so that they were processed by different programs, they would still be two files, although they would be in one pack of cards.

The General Purpose Scientific Document Code system is designed to process files of information written in GPSDC on magnetic tape, drum, or some other means of bulk data storage. Under Exec 2 the only permanent means of bulk data storage was magnetic tape. Drum storage was available only for "temporary" or "scratch" use, that is, a drum file was lost at the end of a run, when the FIN card was read. Under Exec 8 random-access (disk or drum) files can be either "temporary" or "permanent". A permanent, or catalogued, file is one that is saved after the run creating it signs off. The programmer may continue to save it for two days or two years. When it is no longer needed, it can be deleted, or unsaved.

Exec 8 provides not only for "data files", such as a bibliography in GPSDC or a deck of laboratory measurements, but also for "program files." The permanent, read-only Fastrand files of Exec 2 were program files. Exec 2 also provided a single temporary program file during a run on drum, the PCF, which could be written on, read, from, or manipulated, but which vanished at the end of a run. Under Exec 8 each user is provided with a temporary program file or TPF$. (discussed at greater length later), analogous to the PCF. However he will also be able to create (during a run) any number of program file, either permanent or temporary, with optional protection against overwriting. (He will, of course, have the same flexibility with data files.)

2.1. One might conclude from the above that the control language for Exec 8 must be somewhat more complicated than that of Exec 2, because of the many new options. That is true. But there is a certain basic logic to it which makes it relatively easy to remember once one has understood it.

One difference between Exec 2 and Exec 8 control cards should be noted. In Exec 8 the mnemonic or control code comes immediately after the commercial at sign, @, used as a system control; no space separates them. The options come after the control mnemonic separated from it by a comma. For instance, the FORTRAN compiler card may have the form:

```
@ FOR,N      PROG1,PROG1
```

The letter N is an option. (It suppresses the program listing.)

3. Run Card

A typical run card is as follows:

```
@RUN,M   NAME78,12345-NAMEXY,GPSDC,1,50/200
```

The option, M, in the sequence RUN,M is the priority. This is the standard, lowest priority. Higher priorities, F and J should be used only for testing and in emergencies. The first field, NAME78 is called the run ID. It consists of six characters, the first four of which are the first four letters of the user's last name, and the last two of which may be chosen arbitrarily. It is used by Exec 8 to identify the run in communications with the operator, etc. The second field, 12345-NAMEXY, is the account number. It consists of a five digit number, followed by a hyphen, followed by the user's name truncated to 6 characters. In essence, it combines the first two fields from an Exec 2 RUN card. The five digit numerical portion is assigned by the Computer Services Division; those honored under Exec 2 will continue to be accepted under Exec 8. The account field is used for billing purposes.

The third field, GPSDC, is the "project name". (It has no relation to the budgetary concept of a project or cost center). It may be one to twelve characters. It is used for a variety of purposes, principally in connection with the creating and accessing of files. It has the effect of defining a collection of files which will be readily available to the user. The project name GPSDC will be used for the program files containing the Document Image Code system. Users are therefore advised to sign on with this project if they wish to use the system. This will simplify the control cards. For instance, in the run stream shown earlier, the MAP card was @MAP,N DICX8.ASKIN, ASKIN. If the project on the run card had been, say, IMR, the MAP card would have had to be @MAP,N GPSDC*DICX8.ASKIN,ASKIN.

The remaining three fields on the run card are the time, page, and card limits. This run is allowed a maximum running time of one minute and is limited to fifty pages and two hundred cards. Note that a slash, rather than a comma, precedes the card maximum field. If it is omitted, Exec 8 (like Exec 2) will prohibit punched output.

4. Naming of Files and Elements

The basic unit of storage, as previously stated, is the file. A program file is subdivided into elements. The definition of an element is the same under Exec 8 as under Exec 2. For instance, a Fortran subroutine is an element. So is a MAP.

4.1 The full name of a file consists of two parts, separated by an asterisk. The first is the qualifier. This is one to twelve characters, and is normally the "project" field from the run card of the run that created it. The second part is the file name itself. For instance, the full name of the program file containing the GPSDC system is GPSDC*DICX8.

The qualifier may be omitted in references to files only if it is the same as the project name on the run card. For instance, the run which signed on as

```
@RUN,B    PCHEM1,12345-NAMEXY,GPSDC,1,50/200
```

could correctly reference the file GPSDC*DICX8. simply as DICX8., since GPSDC is on the run card.

In normal usage a file name ends with (is terminated by) a period. There are no examples of file names in this manual where the absence of a period is required. In some of the cases shown, however, the period is mandatory; its omission may cause the run to be aborted.

4.2 The name of an element consists of the name of the file containing it, followed by a period, followed by the specific name of the element. For instance, the full name of the editing program EDBØSS is GPSDC*DICX8.EDBØSS. The qualifier may be omitted, as in DICX8.EDBØSS. The file name may not be omitted unless it is TPF$. An element name may consist of one to twelve characters.

Wherever the statement is made in this report that a name, password, et cetera, may consist of a certain number of characters, alphanumeric characters - letters or digits - should be used. The rules restricting the use of special characters are complex. The safest course is to avoid their use except as punctuation required by Exec 8.

5. Assigning Files

5.1 Examples:

```
Mounting a magnetic tape
@ASG,T     3,8C,9876
Assigning a scratch mass storage file
@ASG,T     37,F///64
Creating a catalogued (saved) mass storage file
@ASG,UP    JUNK.,F///64
@USE       3,JUNK.
Assigning a previously saved file
@ASG,A     JUNK.
@USE       3,JUNK.
```

One aspect of Exec 8 is simpler than in Exec 2. That is the assignment of Fortran logical units to files. The Exec 2 conventions of identifying tapes by single letters, and of predefined relationships between logical units and particular tape or drum files, are not found in Exec 8. Instead, if a file name is a 1 or 2 digit integer, it will be accepted by Fortran as a logical unit for input or output. If the file already has a (permanent) non-numerical name, it is possible to assign it a temporary alternate numerical name with a special control card, the USE card.

There are two principal types of ASG cards. The first is the magnetic tape ASG card. To mount tape reel number 9876 as logical unit 3 (GPSDC file 3), one places the following card in the deck:

```
@ASG,T    3.,8C,9876
```

The T option indicates a temporary assignment: The file is not permanently catalogued. (The reel will however be saved.) The first field is the file name. The 8C in the second field is the device type code: it requests a Uniservo 8C tape drive (All of NBS's tape drives are model 8C.) The code 8C9 requests a 9-track tape. The code 8CB requests hardware BCD translation for a 7-track IBM tape.

This identifies the assignment as one of tape, in which case the third field is the reel number. If no reel number is provided, scratch tape will be assigned. The use of scratch tape is not normally necessary; usually scratch drum can be used instead. (Scratch tape is needed for certain specialty applications or for extremely large files.)

5.2 Word and Sector Addressable Formats

In Exec 8 random-access files (disk, drum, etc) may have either of two formats, word-addressable format or sector-addressable format. A word-addressable drum file may be used only as a temporary data file; it may not be catalogued nor used as a program file. This was the format of all Exec 2 drum data files, and is somewhat more efficient than sector-addressable format. To assign 50000 words of drum as unit 36, one can use the control card:

```
@ASG,T    36.,D/50000
```

The device code D indicates a word-addressable drum file.

A sector-addressable drum file is read and written in 28-word sectors and allocated in 1792-word tracks. To assign 64 tracks of mass storage as unit 37. for a temporary file:

```
@ASG,T    37.,F///64
```

(Note that the number of slashes differs for word-addressable and sector-addressable files.)

5.3. Permanent and Temporary Names. Public Files

The name of a permanently catalogued file should be descriptive and should include alphabetic characters. The USE card can assign it a temporary number, making it accessible to Fortran. To create a permanent 64-track file named JUNK and to assign it as unit 3, the cards are:

```
@ASG,UP     JUNK.,F///64
@USE        3.,JUNK.
```

the U option creates a new permanently catalogued (saved) file. The P option states that the file is to be "public." A file can also be "private"; but since this distinction is of questionable utility, we recommend that all saved files be "public."

The USE card makes the number 3 become a temporary alternate name for the file. The first field on the USE card is the new temporary alternate name; the second is the old (standard) name.

After the file JUNK is catalogued, later runs can refer to it as follows:

```
@ASG,A              JUNK.
@USE                3.,JUNK.
```

The A option requests an existing permanently catalogued file.

5.4. Tape Labelling and Tape Options

```
Assigning an Exec 2 GPSDC tape
@ASG,TJM    4.,8C,0135
Assigning a Digidata tape
@ASG,TJLE   8.,8C,D0135
```

Exec 8 employs what is known as "tape labelling" to protect against accidental overwriting of tape files due to operator or programmer error. A label consists of two or three records preceding the body of a tape. At the beginning of a run the executive system checks the label to determine whether the correct reel has been mounted, et cetera.

However, the tape label is written in a format peculiar to Exec 8. If the tapes was created under Exec 2, or outside NBS, or if it is to be read elsewhere, or printed on the 360 printer, the writing and checking of labels must be suppressed. This is done with the J option on the tape ASG card. Tape labelling is assumed. The J option is required to turn it off and provide compatibility.

Under Exec 2 the normal recording density of tape was 556 bpi. The standard density for Exec 8 is 800 bpi. To read an Exec 2 tape the M (medium density) option is required. (This does not apply to 9-track tapes, nor to tapes written at 800 bpi under Exec 2 by use of the X option.)

A Digidata tape is a special case. It is recorded at low density (200 bpi) and in even parity. To read it properly the L option and the E option are required, as well as the J option to turn off label checking.

6. File Security

```
Creating a file with passwords
@ASG,UP      STUFF/X/YZ.,F///128
Assigning a file with passwords
@ASG,A       STUFF/X/YZ.
```

The security of Exec 8 file can be protected in either of two ways. The preferred technique is the use of security keys, or passwords. These are provided at the time a file is catalogued (saved). Access to a file which has security keys (passwords) will be restricted to users who know the passwords.

Public and Private Files

The other method is the distinction between "public" and "private" files. This distinction is of dubious value. It is recommended that all permanent files be created as "public" files. This is done by providing the P option on the ASG card that creates a permanent file. An example would be @ASG,UP JUNK.,F///64

Read and Write Keys (Passwords)

When a file is permanently catalogued, it can be given either a read key or a write key or both. The key, or password,may be 1 to 6 characters long. If a file has keys they must be provided on the ASG card to gain permission to read or to write as the case may be. To create a public file named STUFF with a read key of X and a write key of YZ, and a maximum size of 128 tracks, the ASG card should be:

```
@ASG,UP    STUFF/X/YZ.,F///128
```

Note the use of slashes. A later run wishing to read from and write on the file would need a card:

```
@ASG,A  STUFF/X/YZ.
```

A file can be assigned with only a read key by omitting the write key. To create a file with a write key but no read key, the proper format is:

```
@ASG,UP    NONSENSE//ABC.
```

Any later run can read from the file NONSENSE. A run wishing to write on it must provide the key:

```
@ASG,A     NONSENSE//ABC.
```

A run attempting to use a protected file without providing the password (s) will be terminated.

The file DICX8 has a write key to protect is programs against inadvertent changes. However, no password is needed to read from it, or to use the programs in it. The write key does not interfere with reading from the file.

The use of read keys is not normally recommended except for sensitive files (e.g., personnel records, classified atomic energy information). The use of a write key, however, is a reasonable security measure for important medium-sized or large files whose accidental destruction would result in excessive delay or expense to restore them.

It must be emphasized that a user who creates a file with keys is responsible for remembering the keys. Loss of file keys can cause long delays and serious inconvenience before the keys can be recovered by the Computer Services Division from Exec 8.

7. Deleting Files from Mass Storage

A charge will be made for the use of mass storage for saving files. Users are therefore advised to unsave, or delete, any files which they previously saved but no longer need. The simplest procedure for deleting a file, for instance, the previously mentioned file JUNK, is

```
@DELETE    JUNK.
```

The DELETE card calls a special processor named FURPUR, which is analogous to Exec 2's CUR, to delete the file and release its storage.

If the file has security keys (passwords), they must be given on the DELETE card. For instance,

```
@DELETE  STUFF/X/YZ.
```

A run attempting to delete a file without providing the keys will be terminated.

8. Compiling and Recompiling Programs

```
Compiling a program from cards
@FOR,IS   JUNK.TEST,JUNK.TEST
(Fortran program)
Recompiling a program, inserting it into the TPF$
@FOR,S    DICX8.NXTL1,NXTL1,NXTL1
(Correction cards)
```

Under Exec 2 it was necessary to load a saved program (e.g., part of the Document Image Code system) into the PCF before it could be recompiled or executed. Under Exec 8 this is not necessary. A processor can reference elements in any file, not just the TPF$.

To compile a program from cards and insert it in the file JUNK with the name of TEST, the run stream could contain:

```
@FOR,IS   JUNK.TEST,JUNK.TEST
      (Fortran program)
```

The I option states that a deck is being inserted from cards. This convention differs from Exec 2, where card input was considered normal. In Exec 8 file input is normal. To recompile the program TEST with corrections, the sequence could be:

```
@FOR,US    JUNK.TEST,JUNK.TEST
 (Corrections)
```

The U option calls for updating an existing program. The S option provides a listing of the course program. Unlike Exec 2, the Exec 8 compiler assumes the N option. The first name is the name of the symbolic program. The second name is that of the relocatable output. In Exec 8 a symbolic element, a relocatable element, and an absolute element may all share the same name. The use of separate versions is not necessary as the system makes the distinction automatically.

8.1. The file DICX8 is catalogued with a write key. Therefore, if elements in it are recompiled with corrections, the updated version cannot be stored on the file, in the manner shown above. Corrections to them must be made by placing the corrected element in the TPF$. This can be done as follows:

```
@FOR,S  DICX8.NXTL1,NXTL1,NXTL1
 (Correction cards)
```

The first name is the input source element. The second name is the relocatable output. The third name is the updated source element. (The order of the last two names has been reversed from Exec 2). The third name may be omitted, in which case no source output will be produced. Source output is seldom necessary if the output is to the TPF$. The omission of file names causes both output elements to be placed in the TPF$. In this case neither the I nor the U option is used. When an absolute program is generated by the MAP processor, the TPF$ will be searched before any other file.

8.2. A procedure similar to that shown above for DICX8 programs can be used when a programmer wishes to modify a program, creating a different program with a new name. To recompile the program TEST, changing its name to BOMB, one would use the card:

```
@FOR,S  JUNK.TEST,JUNK.BOMB,JUNK.BOMB
 (Correction cards)
```

The first name is that of the input source program. The second name is that of the relocatable output. The third name is the updated source element. Both new elements are placed in the file JUNK. Once again, the second and third names are reversed from Exec 2).

9. Executing Programs

Mapping and executing a main program

```
@MAP,N      DICX8.ASKIN,ASKIN
@XQT        ASKIN
```

The only kind of main program that can be executed under Exec 8 in the GPSDC system is an absolute program. An absolute program is produced by a special processor which collects the main program, all the subroutines, and any library routines (e.g., NTRAN) into a single element. In Exec 2 this was done automatically, in response to the XQT card. (It could also be done separately via the ABS card.) In Exec 8 an absolute element must be generated before the XQT can be performed. A control card, the MAP card, is used to generate an absolute program. (Sophisticated users of Exec 2 will see that the Exec 8 MAP card combines the functions of the MAP card and the ABS card. Exec 8 does not have "processed map" elements.)

To generate an absolute program and execute it, a MAP card is used, followed by an XQT card. For instance, to run the GPSDC main program ASKIN, the sequence is:

```
@MAP,N    DICX8.ASKIN,ASKIN
@XQT      ASKIN
```

This form assumes that a special symbolic element, the map, has been included in DICX8 by those responsible for its maintenance. The first name on the MAP card is that of this element. The second name will be given to the absolute element. Notice that the absolute element will be placed in the TPF$. The N option suppresses a lengthy memory allocation listing. Options are not used on the XQT card.

It is not possible to execute a main program in the Exec 8 GPSDC system without first using a MAP. The file DICX8 contains relocatable elements of each routine and symbolic MAPS for each main program. It does not contain absolute versions of each program. This procedure has been adopted in order to assure that the current versions of utility routines are always used and to reduce the size of the program file. In the example above the MAP and the absolute element it created have been given the same name. This is the normal practice for GPSDC programs.

Neither the MAP card without a name, nor the XQT without a name can be used with the GPSDC system. Both of these usages require that there be only one main program; GPSDC has several. They also assume that all elements will be in the TPF$, rather than in a permanent file such as DICX8.

10. Writing a MAP

A programmer who has compiled a main program which calls routines in the file DICX8 and wishes to execute it must provide a map. The simplest case is when the main program, named perhaps MAIN, is in the TPF$. The minimum control sequence needed is:

```
@FØR,IS       MAIN,MAIN
(Fortran program)
@MAP,IS       ZOO,ZOO
  IN          MAIN
  LIB         GPSDC*DICX8.,TEXTPROCESS*LIB.
  IN          IO
@XQT          ZOO
```

The I option on the MAP card indicates that the MAP is coming in from cards. The IN card provides that the element MAIN and all elements called by it will be included in the absolute program. The LIB directive tells the MAP processor to search the file DICX8 for any routines that are called but are not in the TPF$. The file TEXTPROCESS*LIB., which contains needed utility and system-oriented subroutines, is also searched. The element IO, which is a "non-standard NTAB$ table", must be included for GPSDC to run properly. (The S option on the MAP card causes a memory allocation listing to be printed.). Since a charge will be made for core storage used, the cost of a run can be reduced if overlaying is provided. This requires a more complex map. Information on overlay maps is available from the Computer Services Division.

11. Sample Run Streams Using Features of Exec 8

The three run streams shown below illustrate the use of many of the features of Exec 8 GPSDC. First DURIN is used to read a Digidata tape and create a saved GPSDC file with read and write protection. Then EDBØSS is used to edit the saved file, creating a new saved file with write protection. Then DGMDMP is used to produce a tape which will drive the 1403 printer.* The routine EDIT, used by EDBØSS, must be corrected.

*At this time the unedited GPSDC file is deleted.

```
@RUN                    NAME71,12345-NAMEXY,GPSDC,3,150
@ASG,TLEJ               8.,8C,9753
@ASG,UP                 GARBAGE/NBS/USA.,F///64
@USE                    1.,GARBAGE.
@MAP,N                  DICX8.DURIN,DURIN
@XQT                    DURIN
*ØFILE=1
FILE 1 NEW
*RM=85
*TABS                   10,20,30,55,70
*LF 1=2
*PGLENG 1=120
*RUN
@EØF
@FIN
@RUN                    NAME72,12345-NAMEXY,GPSDC,2,40
@ASG,A                  GARBAGE/NBS.
@USE                    1.,GARBAGE.
@ASG,UP                 NEWFILE//WØRLD.,F///64
@USE                    4.,NEWFILE.
@FØR,S                  DICX8.EDIT,EDIT,EDIT
(Correction cards)
@MAP,N                  DICX8.EDBØSS,EDBØSS
@XQT                    EDBØSS
FILE 1 ØLD
FILE 4 NEW
*LF   1=2   *LF 4=2
*FØLLØW
*PGMØPT=16
*DØMFIL=1
*RUN
(Edit cards)
@EØF
@FIN
@RUN                    NAME74,12345-NAMEXY,GPSDC,4,25
@ASG,A                  NEWFILE.
@USE                    3.,NEWFILE.
@ASG,TJ                 11,8C,8901
@DELETE                 GARBAGE/NBS/USA.
@MAP,N                  DICX8.DGMDMP,DGMDMP
@XQT                    DGMDMP
*INFILE=3
FILE 3 ØLD
*RUN
@EØF
@EØF
@FIN
```

12. Summary: A Few Rules for Using the GPSDC System

Include the project name GPSDC on the RUN card
Provide the FORTRAN logical unit number on the ASG card or USE card
Write disk files where practical and appropriate ($\leq$ 1 week storage)
Write only public files
Include security keys (passwords) on the ASG card
Correct or modify routines only into the TPF$ (The GPSDC file is protected with a write key)
Map each main program before executing it
The name of the GPSDC program file is DICX8
Use the J option on the magnetic tape ASG card for an Exec 2 or non-NBS tape, or a 360 printer tape

13. Questions and Answers

Q: My run terminated with the message "I/O ERROR 22". What does this mean?

A: You attempted to write beyond the limits of a mass storage file. One possible cause is a loop in a program. If this is not the problem, then the ASG card must be changed to make the file larger.

Q: What rule can be used to decide how many words of drum or tracks of Fastrand to request for a file?

A: At the end of a GPSDC run, a line is printed which gives a count of "blocks" written, as "63 RECORDS WRITTEN IN 10 BLOCKS." One block is 256 words. One track of Fastrand or disk contains approximately 6-1/2 blocks. (It would contain exactly 7 blocks except for wasted space.) When one is converting paper tape to GPSDC, one should allow 1 block (256 words) for approximately 4 feet of paper tape, or 1 track for 25 feet.

Q: What routines are in the file DICX8?

A: The file GPSDC*DICX8 contains all of the programs which were formerly in any of the following Exec 2 files: DOCIM3, DOCSP1, GPSOUT, RFMS1, LINO, SORDC, CKIC11, ASCOUT. All of these files have been merged into DICX8.

Q: How long can a filename be?

A: A filename may be 1 to 12 characters. However, a Fortran program (the GPSDC routines are in Fortran) can refer to a file only by a number. This number may be the file's name, or it may have been attached to it via the USE card.

Q: My run terminated with the message "I/O ERROR 20". What does this mean?

A: You tried to use a protected file without providing the necessary keys (passwords).

Q: May I modify a routine that is in DICX8?

A: Yes, but. You may modify it placing a new copy in the TPF$., but not in DICX8, since that file is protected by a write key.

You may modify a routine only if its symbolic element is in DICX8. In order to save mass storage space DICX8. has only relocatable elements for most of the standard routines. The symbolic elements are on a backup tape.

FORM NBS-114A (1-71)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBSIR 73-198	2. Gov't Accession No.	3. Recipient's Accession No.

4. TITLE AND SUBTITLE

General Purpose Scientific Document Code Operations under Exec 8

5. Publication Date

6. Performing Organization Code

7. AUTHOR(S)

Robert McClenon

8. Performing Organization NBSIR 73-198

9. PERFORMING ORGANIZATION NAME AND ADDRESS

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

10. Project/Task/Work Unit No.

11. Contract/Grant No.

12. Sponsoring Organization Name and Address

Same as 9 above

13. Type of Report & Period Covered

Interim Report

14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

16. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)

Operating procedures for the use of the General Purpose Scientific Document Code text handling system are described. These apply to the use of the system on the NBS 1108 computer under control of the Exec 8 supervisor. Control cards, file definition and handling, compilation of routines and execution of programs are described. Examples of typical runs are given. The report is a manual for users of the system.

17. KEY WORDS (Alphabetical order, separated by semicolons) Character codes for scientific text; Computerized text processing; NBS computer system; Scientific text; Users manual

18. AVAILABILITY STATEMENT

[X] UNLIMITED.

[ ] FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NTIS.

19. SECURITY CLASS (THIS REPORT) UNCLASSIFIED

21. NO. OF PAGES

20. SECURITY CLASS (THIS PAGE) UNCLASSIFIED

22. Price

USCOMM-DC 66244-P71